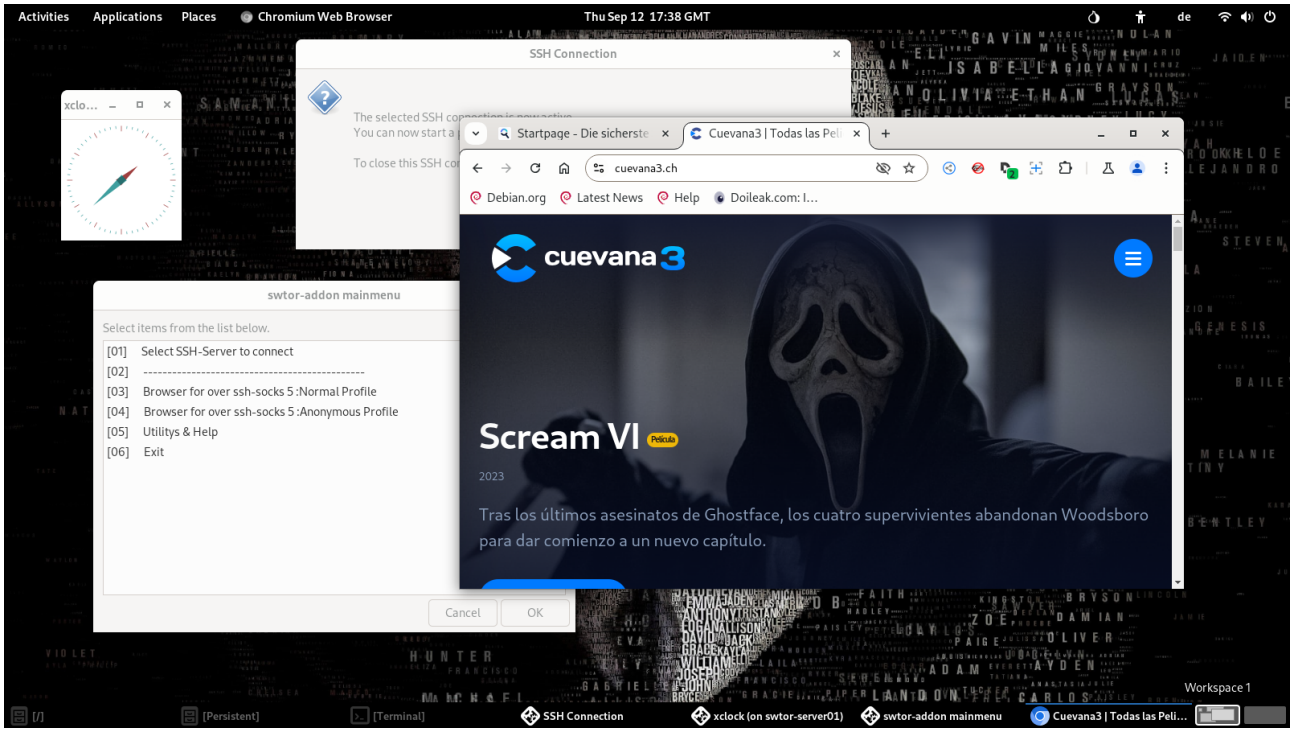
Documentation for the add-on to Tails

Author	:	swtor00@protonmail.com
Date	:	21.11.2025
Version	:	Release 0.90 / needs Tails 7.2 or higher
Licence	:	GPL 2.0



In 2017, I began writing the first iteration of this script strictly for my personal use. This initial implementation was a substantial shell script and ran in the background with Tails 2.2. Its primary purpose was not to browse Tor-unfriendly sites.
I was using Tails almost every day, and whenever a customer called, I had to shut it down and boot my installed Windows system, where all the customer VPN configurations were stored. Eventually, I grew tired of rebooting every half an hour, only to work for a few minutes and then reboot into Tails again. With an outgoing SSH connection, I could assist my customers (via VNC Viewer or rdesktop for Linux) even while directly within a Tails session. It was at this point that I began to realize this rather crude script could also be used to visit multiple websites without encountering blocked services or the frustration of "captcha terror," precisely because I was using Tails or TOR-Browser.

This initial shell script grew rapidly, quickly becoming a small mess that was difficult to extend and maintain. And yes, it was ugly to use and not well integrated into Tails. Prior to version 3.8 of Tails, installing software via the GUI was not possible; I could only implement this feature using the terminal. Consequently, my first script was entirely terminal-based, presented without any fancy graphical user interface. The source code was not openly available to the public until 2018, when I published it under a free license on GitHub.

The current version 0.90 of the add-on has made the 'monster script' I started with version 0.0.1 a lot more user-friendly. Here’s what you get:

- A GUI menu system that even beginners to Tails or Linux can handle.
- Better integration into Tails than ever before.
- Since version 0.83, the add-on can even auto start and initialize with activated DOT-files.
- The ability to build a local SOCKS5 server and make an SSH connection to a foreign host via the Onion Network. This neat trick lets you visit Tor-unfriendly sites while running Tails.
- Easy backup and restore of a complete persistent volume on your Tails USB stick.
- Upgrades on the fly using Git.

I hope you have fun using my little add-on to improve your experience with Tails.
Best regards from Switzerland,
swtor00

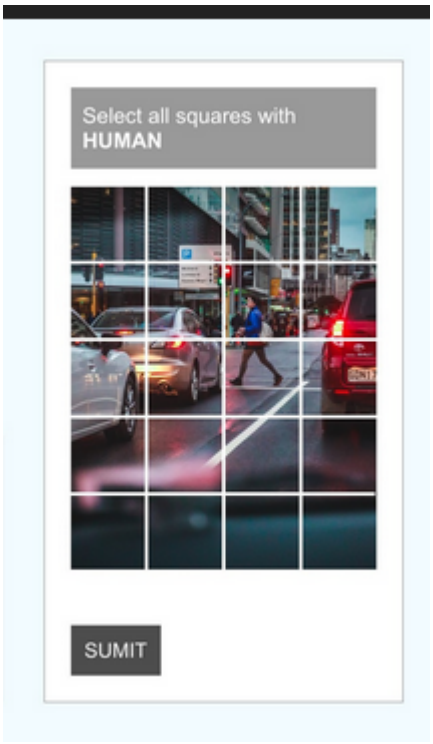
1.0 Introduction

This documentation explains how to install and use this add-on for Tails Linux. You might ask yourself why you need an add-on for Tails in the first place. The Tails system already protects your privacy by using the Tor (Onion) Network on every startup. Tails does a very good job of hiding your true identity and your real WAN IP address from the websites you visit. Honestly, the Tails OS developers and their supporters do this much better than a single user ever could by just installing the Tor Browser bundle on a regular Windows or Linux system. To be ironic: maybe the only real purpose for Tor Browser for Windows is to download Tails and set up the USB stick.

Today, in 2025, many Tor Browser and Tails users still have difficulty navigating the regular World Wide Web (sometimes called the Clearnet Internet —like Google and Facebook, for example). This is because some websites have rules that discriminate against Tor users. When we use Tails or the Tor Browser, our IP packets are sent through three different nodes over the Internet to hide where they came from. The Onion Network's "Exit Nodes" are easy to detect instantly as soon as someone queries a remote IP address from the URL below.

https://check.torproject.org/torbulkexitlist?ip=1.1.1.1

So, any website you visit knows you are using the Tor Browser or Tails, and they will often subject regular users to this act of brainless 'captcha terror'. Below, you can see an example of this 'captcha terror'.



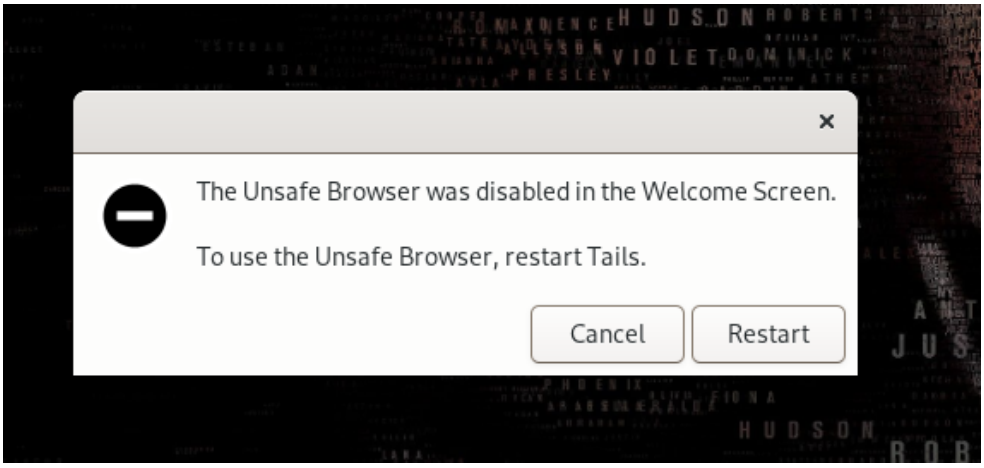
Alarmingly, the number of websites that completely block or implement 'captcha terror' against regular Tor users is increasing daily, significantly diminishing Tor's overall usefulness.

Every Tails installation on a CD or USB stick still includes the 'Unsafe Browser'. This browser doesn't use for the communication the Tor (Onion) Network that Tails runs in the background. As a result, all its data communications can easily be logged by your ISP, your government, or a network administrator (e.g., at a hotel). The 'Unsafe Browser' was actually disabled after Tails version 4.8 came out, but was activated again by default with the release of Tails 6.0.

It probably only makes sense to use the 'Unsafe Browser' in Tails solely to access captive portals (the login pages in hotels or public Wi-Fi networks), but not for anything beyond this simple task. The following URL provides additional useful information about the 'Unsafe Browser'.

https://tails.net/contribute/design/Unsafe_Browser/

As I said, the 'Unsafe Browser' is activated again as of Tails 6.0. I strongly recommend that you disable the 'Unsafe Browser' startup option in the Tails Greeter screen. Make sure to enable the useful setting that displays a warning window whenever you try to start the 'Unsafe Browser'.



- Please do not activate the unsafe browser of tails in the welcome screen of tails if you do not need it, and do not use it in any situation.
- When you visit an external website anywhere on the internet while using the unsafe-tails browser, your actual IP address of your currently used WAN interface will be immediately leaked to the remote site !

Smart people even try to start a VPN connection with OpenVPN or something similar to hide the fact that they are using the onion network, which can be blocked by any website or even an ISP. Apart from the fact that you are only able to use a TCP port for a possible VPN communication (all UDP ports including ICMP messages inside of Tails are blocked by default), it creates many more problems than it would solve. In the endless debate in multiple forums on the Internet about the use of a Virtual Private Network (VPN) inside of Tails, I recommend to read the following URL completely.

https://gitlab.tails.boum.org/tails/blueprints/-/wikis/VPN-support

The developers of Tails (and they know Tails from the ground up with every little detail that other so called technical experts don't even know yet or ever have learned) have a loud and clear statement on their opposition to integrating any kind of a software like Open VPN into Tails Operating System. The only clean and acceptable way for the many developers of Tails to have a fixed outgoing IP address (that is not part of the onion network, of course) is to create a local SOCKS5 server and to establish a SSH connection to a remote host. Now, exactly right here, my special add-on for Tails enters the game called "Enhancing Tails with a kind of VPN," and provides some very useful functions for the many Tails users out there.

- Use of an encrypted SSH connection to a remote host and building of a local SOCKS5 proxy. Even the traffic that is sent over the so called 'exit node' of our communication is still encrypted until it reaches the destination SSH server. When the connection packets leave the SSH server to any external website for example, the packets are no longer protected by SSH itself and would look like any normal network traffic from a standard desktop computer running Linux.

- All SSH traffic is encrypted and routed over the onion network, as long as you are using an external SSH server anywhere on the Internet. If you are using an SSH server on your local network at home, only the connection from your Tails system to the internal SSH server is encrypted. And I assume that this is not the way you would like to go about hiding the fact that you are using Tails.

- A local installed Browser (Chromium) with three different profiles that can be used to visit Tor-unfriendly websites like youtube.com and many others that would block regular Tor users (treating them like second-class internet users). All three Chromium profiles are protected against actions like WebRTC and other trackers in general.

- For any particular website that we are visiting with Chromium on the Internet over the script, it is no longer possible to detect that we are using the onion network to hide our personal information, or, more importantly, our current public IP address. All the traffic the owner of the website can analyze is coming from the regular public IP address of the remote SSH server we are currently connected to.

One significant problem still remains with using the Tails system to contact onion-unfriendly systems at all. The type of working bypass protocol used in the background makes no difference. These 3 bypass systems can be implemented using one of the following three techniques.

1. Using SSH to build a local socks5 Proxy

This is exactly the way this add-on works internal and is the preferred method by Tails developers. The add-on itself uses only already installed Tails software to create our secure SSH connection. For the password less login to a remote SSH-Host we have to install additional software (sshpass).

2. OpenVPN Server with a single TCP port

OpenVPN is a very popular software for creating any kind of VPN. Most public OpenVPN servers on the Internet (there are thousands of them on the net) only work with a single UDP port for connection. The UDP protocol is much faster than the TCP protocol. This is one of the main reasons why so many OpenVPN servers only provide UDP ports. There are not so many OpenVPN servers out there on the Internet that can be used with the TCP protocol. Please take a closer look at the OpenVPN servers on the following URL and you'll see what I mean. Almost 98% of all servers only provide UDP ports for connecting.

https://www.vpngate.net/en/

This very accurate und updated list shows multiple free OpenVPN servers worldwide. I use this impressive list of free OpenVPN servers very often to watch live TV and read some newspapers from my current location while I am on vacation, because every connection attempt from a foreign country is blocked inside my own country where I live. I use this list very often with an OpenVPN client under Linux or even Windows. But not within Tails, for several understandable reasons.
It is one thing just to watch a harmless TV show over a public VPN server that is managed by a person I do not know personally and logs all my connections on that server, but it is a completely different story to send any kind of sensitive data to a server that can be trusted 100%.

3. Install proxychains

During my first experiments with Tails to establish a stable connection to a few Tor-unfriendly websites, I was using this tool called proxychains. Sometimes it worked with mixed results, but half an hour later the program stopped working without a clear reason. In my humble opinion, it wasn't the right tool to do the job properly. You may test it for yourself; you may come to a different conclusion than I have.

The next following three commonly used VPN protocols do not work inside Tails.

- LL2TP
- IPSec
- WireGuard

All 3 of the above-listed well-known protocols use or depend on one or multiple UDP ports to work properly. Inside Tails, it is not possible to create a simple UDP connection directly to the internet.
A big warning about using any of the three working presented bypass techniques (including the remote SSH servers used with this add-on):

At this point, I would like to raise awareness about the trust you are willing to give to a foreign host system and its administrators or even users, who could easily read your complete communication that would be sent through to the foreign server over Tails. If you would like to use any of the three working bypass techniques, please do not underestimate the control they have over you at that exact moment you are using their servers.

Almost any VPN or SSH service provider (commercial or free) worldwide makes claims on their websites with misleading marketing statements like the following ones:

- “We don’t log anything!”
- “We don’t spy our users!”
- “We protect the privacy of our customers!”
- “You can trust our VPN services!”

To be honest, that is in most cases only cheap marketing intended for foolish customers who truly believe the material published by the company offering the service. Several interesting articles discussing the 'no-log policy' claims of some major worldwide VPN providers can be found right here.

https://www.theregister.com/2011/09/26/hidemyass_lulzsec_controversy/
https://www.pcmag.com/news/7-vpn-services-found-recording-user-logs-despite-no-log-pledge

If you read the above article carefully, you may come to the final conclusion that you should not give your personal trust to the first person or company that offers you a VPN or a full SSH account for free. Even if you pay a monthly fee for a service, there is no guarantee that you are not being 'tracked' by this VPN server or any other active users of the remote system. Some very shameless VPN providers make a second business selling the collected VPN data to other companies for marketing purposes. The company Avast, as a horrible example, was selling customer VPN data over the years!

If we are talking about the trust you are willing to give to a company or a remote server, we should also talk about one of the most secure email providers on the internet called Proton Mail that resides in Geneva, Switzerland. Until 2021, the company Proton also claimed that no log files were generated. The French customer of Proton Mail who was arrested in 2021 by the police may see it completely differently. He was using his free Proton Mail account over the normal https URL, rather than the onion address that Proton provides. If this French customer had used the Tor Browser to access his mail, he may not have been arrested.

https://account.protonmail.com

This was the normal URL of Proton Mail he used prior to his arrest. Proton gave the French police department the public WAN IP address he was using to access his personal mail. A few hours later, he was arrested and in handcuffs. Remember this: when a company says publicly what they would never do, they will do it for sure if they have to. If you are using ProtonMail (as do I), my personal advice is clear: Only use the Proton onion v3 address. Only use this save URL and never the Clearnet URL

https://protonmailrmez3lotccipshtkleegetolb73fuirgj7r4o4vfu7ozyd.onion/login

This means you should do this only with Tails. And do not under any possible circumstances check your private Proton Mail account with a standard PC operating system like Windows outside of Tails. The moment you read your email outside of the Tor network, however, your public IP address is visible to the company Proton. Prior to showing three possible working SSH scenarios with the add-on, we have to talk about the dangers of using them at all. Of all the three SSH scenarios described in the next chapter, you should only use the last presented scenario (Scenario 03) whenever possible.

- Never create a login with a username/email address/password that you have ever used on the standard internet anytime.

- A special note about the above rule, if you are living in a country like China or something similar that completely blocks Google mail. For example, if you are the owner of a Gmail account, you would never have a real chance to login over the Tor Browser. With the help of the add-on, it would be possible to get a chance to login for that particular Gmail account. That’s maybe the only possible exception! If you don’t need access to a Gmail account while you are using Tails, stay away from Google.

- If you need a valid email address to register for a service or something similar while using Tails, you must create a single email address that you use only inside of Tails and nowhere else.

  Personal tip : Use this URL to create a anonymous email

  http://hxuzjtocnzvv5g2rtg2bhwkcbupmk7rclb6lly3fo4tvqkk5oyrv3nid.onion/

- Never try to access or publish this specially created email login to anyone outside of Tails !

- Never send an email from your ordinary email to this Tails-only generated email address !

- Never send an email from the Tails-only email to anyone that you ever contacted by your ordinary email system at home or in a business !

- These kinds of digital fingerprints (emails/nicknames/passwords and many more little things that you often do not think about) could be easily tracked back to you as a person, given enough time and effort. Never use an email address for a possible password reset in the future that directly points to you !

- Only use the add-on to solve problems with Tor-unfriendly sites, and if possible, use the more secure built-in Tor Browser of Tails every time to connect to them.

- You should never use a personal credit card to pay anything, which points directly to you.

- You should never use any kind of two-factor authentication that requires a mobile number for registration or login via SMS. It would be possible to use 'Google Authenticator' on Linux, which also works offline. Installation and use of this application is outside the scope of this manual.

- Never use any websites simultaneously inside the Tor Browser and the Chromium Browser used by the script. This is because if your internet connection drops, both of your connections will terminate at the same time, and it will not be hard for someone spying on you to recover the pieces and complete the puzzle. It may be better to use only one browser at any given time and not try to mix them up.

- We are already in the year 2025 and we still have millions of dummy websites that still use the outdated and unsecure HTTP protocol instead of the secure alternative HTTPS. You may know that Tor can be exploited using vulnerabilities present at its exit nodes. So, if you access HTTP sites using Tor, there are chances someone might access your information while it is at the exit node. Data transferred to and from an HTTP site is not encrypted and can be viewed at the exit node, as Tor only encrypts traffic within its own network. You can prevent such situations by using HTTPS websites. HTTPS websites use transport layer security protocols like SSL and TLS. So, all your data remains safe, even when it is outside the Tor network to the final destination.

- Never wait more than a few days to update the Tails system, if possible. Ensure you always use the latest stable release of Tails.

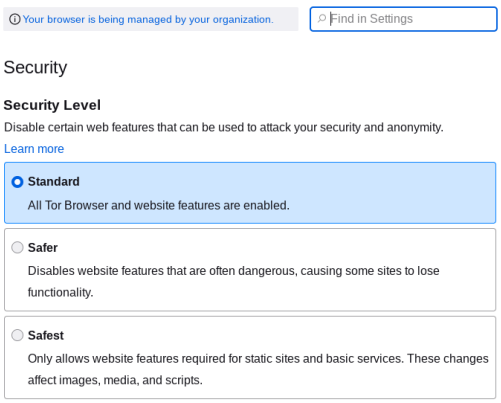
- Only use search engines inside both browsers that do not store personal settings of any kind. There are two known public search engines that do this; they do not store anything about your searches, and as of 2025 these two search engines have not been caught doing so.

  https://www.startpage.com
  https://duckduckgo.com/

- Do not use and activate the unsafe browser of Tails. (Yes, I know this is the third warning I have given you.)

- Do not try to install any P2P software, such as BitTorrent clients, within Tails.

- Some people are suggesting an increased security level (from Standard to Safest) for the Tor Browser that can be set by the user anytime.



Warning: If you increase the level to the highest mode 'Safest,' almost any modern website is looking horribly ugly and is more or less useless. I never increased this level during the long time I used Tails.

- If you are downloading any kind of files with any of the two browsers inside Tails (like any Microsoft Office files as a simple example), please don't try to copy them to USB and open them in another operating system like Windows with an installed Office suite in place. Files downloaded with Tails should be opened only within Tails and never leave the system.

- What happens in Tails → remains in Tails.

https://www.thedailybeast.com/this-is-how-cops-trick-dark-web-drug-dealers-into-unmasking-themselves

Some very stupid dutch drug dealers downloaded an Excel sheet from the Tor Browser and opened this police-manipulated Excel sheet with their regularly used Windows Computer. The moment they opened the sheet, their public IP was transmitted directly to the police with the help of Office macros.

- Do not try to install the Tor Browser on an ordinary operating system like Windows or Linux and use the same credentials you created already for Tails-only use. If you really have to go this highly discouraged way, then you have to create new email addresses and SSH accounts for every system.

- Save the data from the persistent volume of Tails at regular intervals. This backup should be placed in a very secure location inside your own residence or, even better, transferred securely using SSH to a remote host in a country other than your current one. Using SSH, you can copy files to a remote host as well. When you need this backup in case of a damaged stick (some call it a Tails emergency), you need the following information to copy the backup back to a new Tails stick.

  - DNS name or IP address of the backup server
  - Connection port
  - User name and password
  - Exact location of the backup inside the remote user directory
  - You may also need the SSH keys if you cannot log in with a password to the remote host.

  - Use a strong and long password for the persistent volume of Tails. Please take some advice from me: Use a long password that you can also type with the standard US keyboard layout, or you will have to switch the keyboard first before typing the password for your persistent volume.

- Always try to shut down the Tails OS properly. If you do not shut down Tails properly, the persistent volume could be damaged.

- Never try to visit any Tor-related websites with a normal browser using a Windows or Linux system. This is especially true for the following websites:

  - https://tails.boum.org
  - https://www.torproject.org/
  - https://gitlab.tails.boum.org/tails
  - https://github.com/swtor00/swtor-addon-to-tails
  - https://www.reddit.com/r/tails/

- You know what the funny thing is about the Tails forum on reddit.com ? There are a lot of active Tails users on that forum that try to be as 'anonymous' as possible as long as they are using Tails, yet the first thing they do is this: they register for a new Reddit account with a private, or even worse, a business email address for posting on that forum. This is done, of course, with an ordinary browser on a Windows or Linux system. If you would like to visit these websites, please do it the right way and use Tails. Since the new release 7.2 (with Tor Browser 15.01) it is almost impossible to use reddit.com with the TOR-Browser. You may could use the addon to visit reddit ?

Or maybe this one, as a last very ironic example from me of how you should not post a photo on Reddit: You have a strange error message inside of Tails and you don't know what to do with this message? You take a photo of the error message with your current smartphone and post this photo directly to the Tails forum on Reddit. There are so many nice users of Tails in that forum, perhaps someone knows the solution to your problem. You are probably not aware of this, but the following worst-case scenario can happen very quickly.

- The serial and IMEI number of your smartphone are hidden in the metadata of the photo.
- The exact location measured with GPS may also be included in the metadata.
- And last but not least, the same photo may already be stored in a Google or Apple cloud, depending on the operating system of the smartphone used.

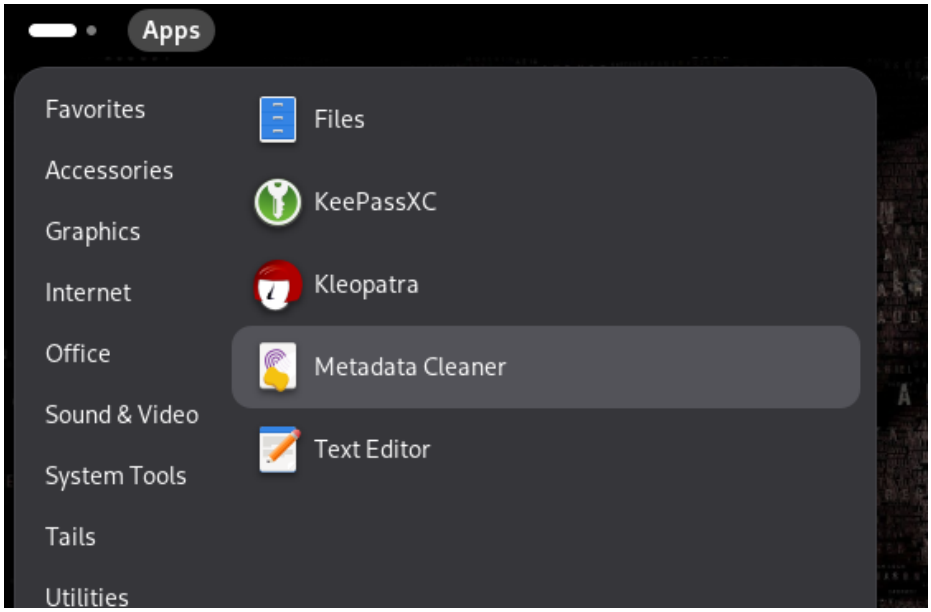
Please do not underestimate the amount of information that could be hidden in the metadata of the following file types (only a few examples).

- Photos with multiple types of information as described above.
- Generated PDF's that contain the serial number or the Adobe Live ID.
- Office documents may contain a serial number/company name or even a real Microsoft Live ID.

Below you'll find a very good overview with many examples about the real danger of metadata inside photos, PDF files, or Word,Excel documents:

https://www.idox.ai/blog/removing-metadata

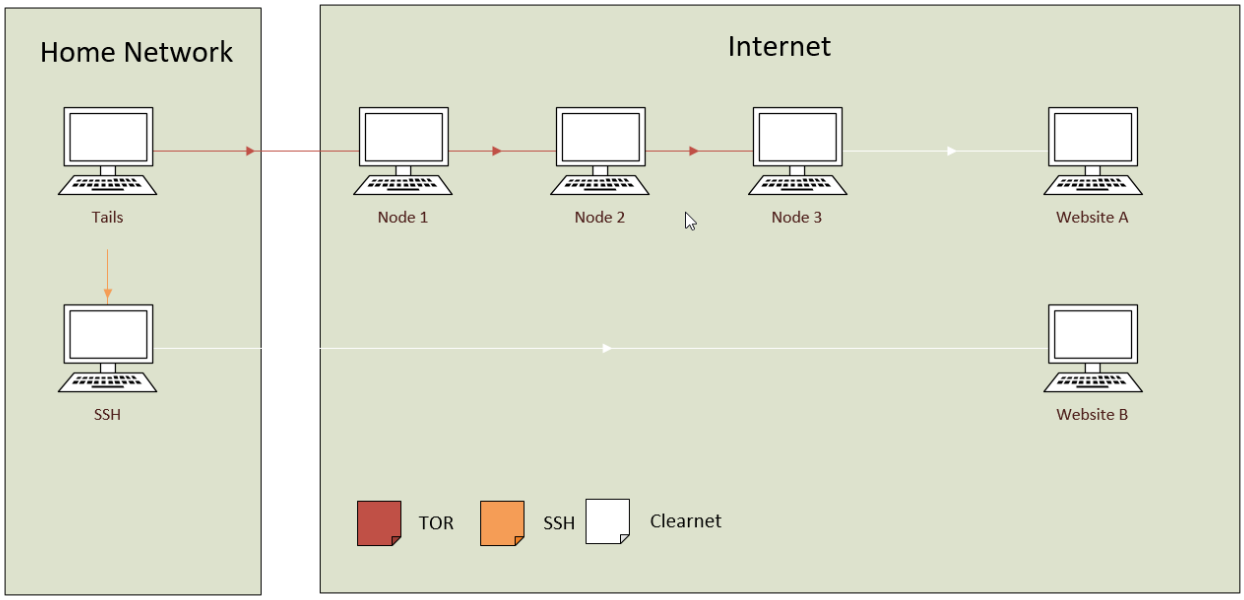
Tails includes a small tool called 'Metadata Cleaner' to remove unwanted metadata from multiple types of files. You can find it under the Accessories menu. You may find all of my above security tips for using Tails in a secure way are way too ridiculous or excessive, but if you are living in a 'Big Brother' country like China with very strong internet censorship, these tips and tricks may help you to stay free rather than being recognized and identified as a Tails user with all the possible consequences.


Apps
Favorites
Accessories
Graphics
Internet
Office
Sound & Video
System Tools
Tails
Utilities
Files
KeePassXC
Kleopatra
Metadata Cleaner
Text Editor

2.0 Using your own SSH server inside your own Network at home

For this simple and not really recommend scenario, you need at minimum, a second computer with Linux or Unix running on your own network at home. For this SSH-daemon you could use, for example, a simple Raspberry-Pi, or of course, any other computer with an SSH daemon would work as well. This could be implemented with a Linux System like Debian or many others without any problem. For a simple example to build a standalone SSH-server on a Raspberry-Pi I would recommend the following URL.

https://phoenixnap.com/kb/enable-ssh-raspberry-pi



This script allows access to sites like Google that usually block Tails connections.

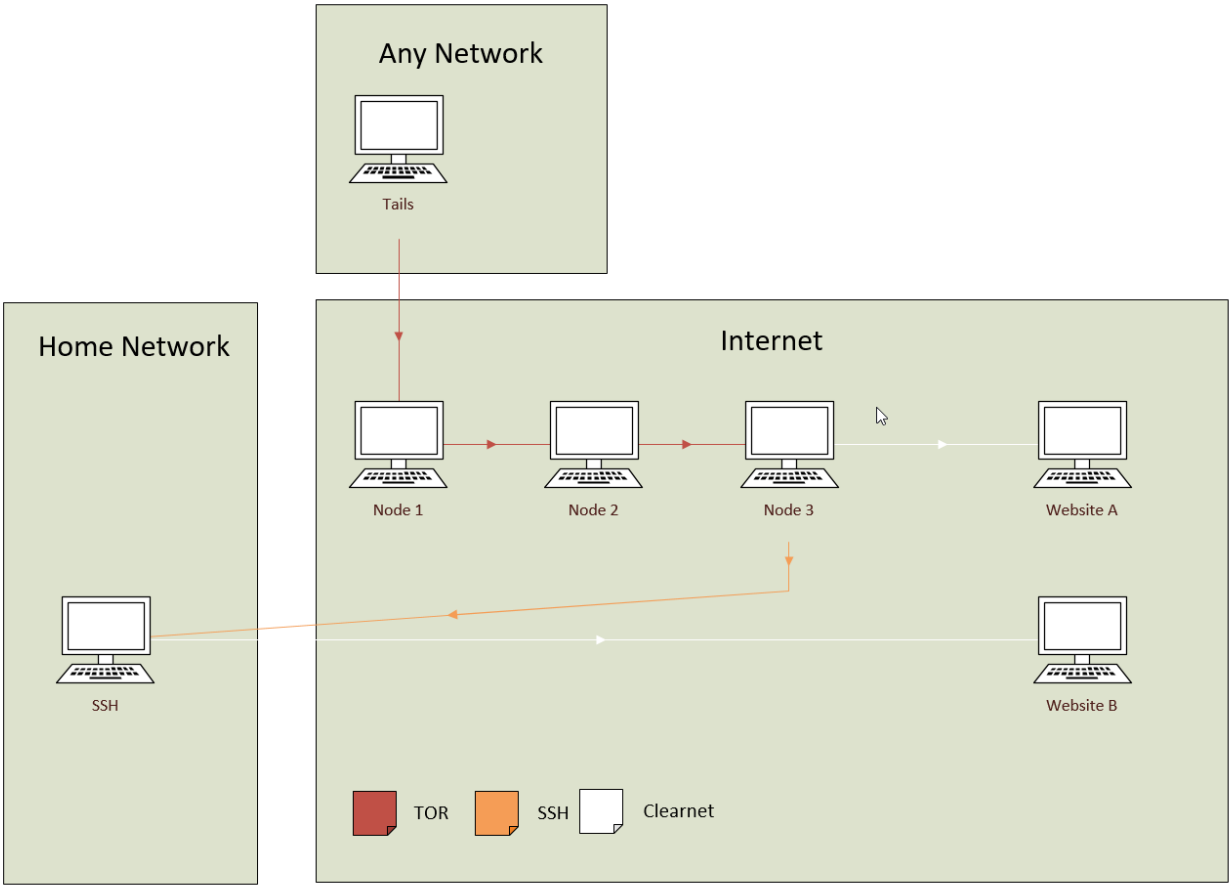
WARNING: Unencrypted Traffic

All internet traffic sent from the SSH-Server via this script in this scenario can be monitored by your ISP. Data flowing from your home SSH server to "Website B" (the white line in the graphic) looks like normal home network traffic and is not anonymous. Only traffic through the Tor Browser ("Website A," the red line) is secure and private.

- Because the destination SSH server resides within the same local physical network, the data you send from Tails to that server remains local; it does not traverse your main router toward the public internet. This traffic is not routed through Tor's three external nodes because it is confined to your local area network (LAN).Only traffic that originates from the SSH connection after it leaves your local network (i.e., from the SSH server outward to the final destination) is exposed to the internet.

- With the new release of 0.83 (Spring 2025) of the addon for Tails and the new implemented redirection feature it would be possible to send all your local data encrypted over ssh to a remote host. In this particular case your ISP would only see the encrypted traffic to this remote server, but he can not track what your are doing exactly (because of SSH).

3.0 Using your own SSH-server that can be used when you are not at home



Connecting to your personal home SSH server externally from the Internet while using Tails requires additional configuration. You need to configure port forwarding within your local network firewall or router. This step directs incoming traffic from the internet to the correct destination IP address within your home network.

Configuration Steps:

1. Enable Port Forwarding: Forward TCP port 22 (or any other port you may prefer) to the internal IP address of your SSH server.

2. Access Your Device Settings: This configuration is typically done within your home router's admin interface or a dedicated firewall management software.

3. Consult Documentation: Because configuration interfaces vary widely between manufacturers, you will need to refer to your specific device's documentation. You can often find general, step-by-step guides for many manufacturers and models on websites like PortForward.com or by searching your manufacturer's support website.

The moment you make your internal SSH server reachable over the internet via port forwarding, it becomes a target for possible external attacks.

Crucial Step: You must implement essential security hardening measures on the SSH server before you activate port forwarding on your router ! All recommended security settings for the SSH daemon are managed within the configuration file located at /etc/ssh/sshd_config. For detailed guides on how to secure this service, you can follow tutorials like this one from DigitalOcean on SSH Hardening (guides for other Linux distributions are widely available).

Disable Root Logins over SSH

It is a critical security best practice to prevent direct SSH logins using the root account. This mitigates brute-force attacks and improves accountability.To enforce this:

1. Open your SSH daemon configuration file (usually /etc/ssh/sshd_config) using a text editor with administrative privileges (e.g., sudo nano /etc/ssh/sshd_config).

2. Locate the line containing PermitRootLogin and ensure it is set to no:

Restrict SSH Access to Specific Users

For enhanced security, limit SSH access to only the specific user accounts that absolutely require it. This minimizes the attack surface by disabling access for all other system accounts.To implement this:

1. Open the SSH daemon configuration file (e.g., /etc/ssh/sshd_config).

2. Add or modify the AllowUsers directive near the bottom of the file.

    AllowUsers freaky

In this example, only the user named "freaky" will be permitted to establish an SSH connection to the server. You must replace freaky with your own desired username.

3. Save the changes to the file.

4. Restart the SSH service for the new settings to become active (e.g., run sudo systemctl restart ssh on most modern Linux systems).

Change the Default SSH Port

Do not use the standard TCP port 22 for your externally exposed SSH server. While this does not make the connection inherently "more secure" (it's obscurity, not encryption), it significantly reduces the noise from automated internet scanners and bot attacks that primarily target port 22. A good replacement would be TCP port 443 (HTTPS) or 53 (DNS), as these ports are often left open by firewalls in public networks. How to configure a new port:

1. Open your SSH daemon configuration file (typically /etc/ssh/sshd_config).
2. Locate the line that starts with #Port 22 or Port 22.
3. Change the port number to your desired alternative (e.g., 443):

Port 443

4. Ensure that a corresponding port forwarding rule is configured in your router (as described in a previous step) to forward the new port number (e.g., 443) to your SSH server's local IP address.

5. Save the configuration file and restart the SSH service for changes to take effect.

Enforce Key-Based Authentication

The most secure method for logging into an SSH server is using SSH keys, which are far more secure than passwords. Once configured correctly, you should entirely disable password-based logins. Temporary Configuration (During Setup). Initially, you need both methods enabled to ensure you can still log in while you generate and transfer your SSH key:

1. Open the /etc/ssh/sshd_config file.

2. Ensure these lines are present and set as follows:

PubkeyAuthentication yes
PasswordAuthentication yes

The Final, Secure Configuration. Once you have successfully generated your SSH key pair and verified that you can log in using that key, you must disable password authentication entirely:

1. Edit /etc/ssh/sshd_config again.
2. Change the PasswordAuthentication value to no :

PasswordAuthentication no

3. Save the file and restart the SSH service.

Enforce SSH Protocol Version 2

SSH Protocol Version 1 is deprecated, highly insecure, and should never be used. You must ensure your server is configured to only accept connections using the modern, secure Protocol Version 2. How to configure this setting:

1. Open your SSH daemon configuration file (typically /etc/ssh/sshd_config).
2. Add or modify the Protocol line to specify 2 :

Protocol 2

3. Save the file and restart your SSH service for the changes to take effect.

Customize the SSH Login Banner

You can configure a custom message that appears before the login prompt (the "banner"). This can be used to display legal notices, security warnings, or generic messages to potentially mislead automated scanners or unauthorized individuals. To set a custom pre-login banner:

1. Create a banner file: Use a text editor to create a new file (e.g., /etc/issue.net) and add your desired message, which can be simple text or ASCII art.

sudo nano /etc/issue.net

Example content for a /etc/issue.net:

****
This is a restricted access system. All activity is logged.
Unauthorized access is strictly prohibited. Disconnect now.
****

2. Configure the SSH daemon: Open the main SSH configuration file:

sudo nano /etc/ssh/sshd_config

Banner /etc/issue.net

3. Save and restart: Save the changes and restart the SSH service for the new banner to take effect:

sudo systemctl restart ssh
# Or, depending on your system:
# sudo service ssh restart

Note: The banner is displayed before authentication, so anyone attempting to connect will see this message. A separate configuration exists for the "Message Of The Day" (/etc/motd), which is displayed after a successful login.

Schedule SSH Server Availability

To enhance security, you can schedule the SSH daemon (sshd) to run only during specific times when you know you will need access. This minimizes the window of opportunity for attackers.
You can automate the starting and stopping of the SSH service using system scheduling tools like cron or systemd timers.

Example using cron (a simple, reliable method for always-on systems):

You can add entries to the root user's crontab to start and stop the service at specific times. The commands to manage the service generally use systemctl.

```
0 8 * * * systemctl start ssh   # Start SSH at 8:00 AM daily
0 22 * * * systemctl stop ssh  # Stop SSH at 10:00 PM daily
```

For more advanced scheduling, or if you need tasks to run even if the system was offline at the scheduled time, consider using systemd timers.

Maintain System Updates

It is critically important to keep your operating system and all installed software updated to the latest available versions. Since your computer is now directly accessible from the internet (due to the port forwarding configuration), it is exposed to constant scanning and attack attempts.

Action Required:

Regularly apply security patches and software updates to protect against known vulnerabilities. This is your primary defense against exploitation. Consult your operating system's documentation for the correct procedure to update your Linux or your specific distribution.

Ensuring Continuous Availability

Your home computer must remain powered on and connected to the internet at all times if you intend to access the server remotely from any external location or any time. The service will be unavailable whenever the host machine is shut down, asleep, or disconnected.

Utilizing Dynamic DNS (DDNS)

Most Internet Service Providers (ISPs) assign dynamic public IP addresses that change frequently. To ensure reliable remote access to your home server, you need a Dynamic DNS (DDNS) service, which automatically maps a consistent domain name (e.g., myhome.dynu.net) to your home network's current IP address.

Note on Security:
While DDNS is a technical necessity for remote access, using it means your home network services are constantly advertised under a fixed hostname to the entire internet. This increases the importance of following all security hardening steps outlined previously. A reputable and secure DDNS provider is Dynu Systems. You can set up a free account and manage your hostname configuration on their website: https://www.dynu.com/.

Critical Warning: Your ISP Can Track This Traffic

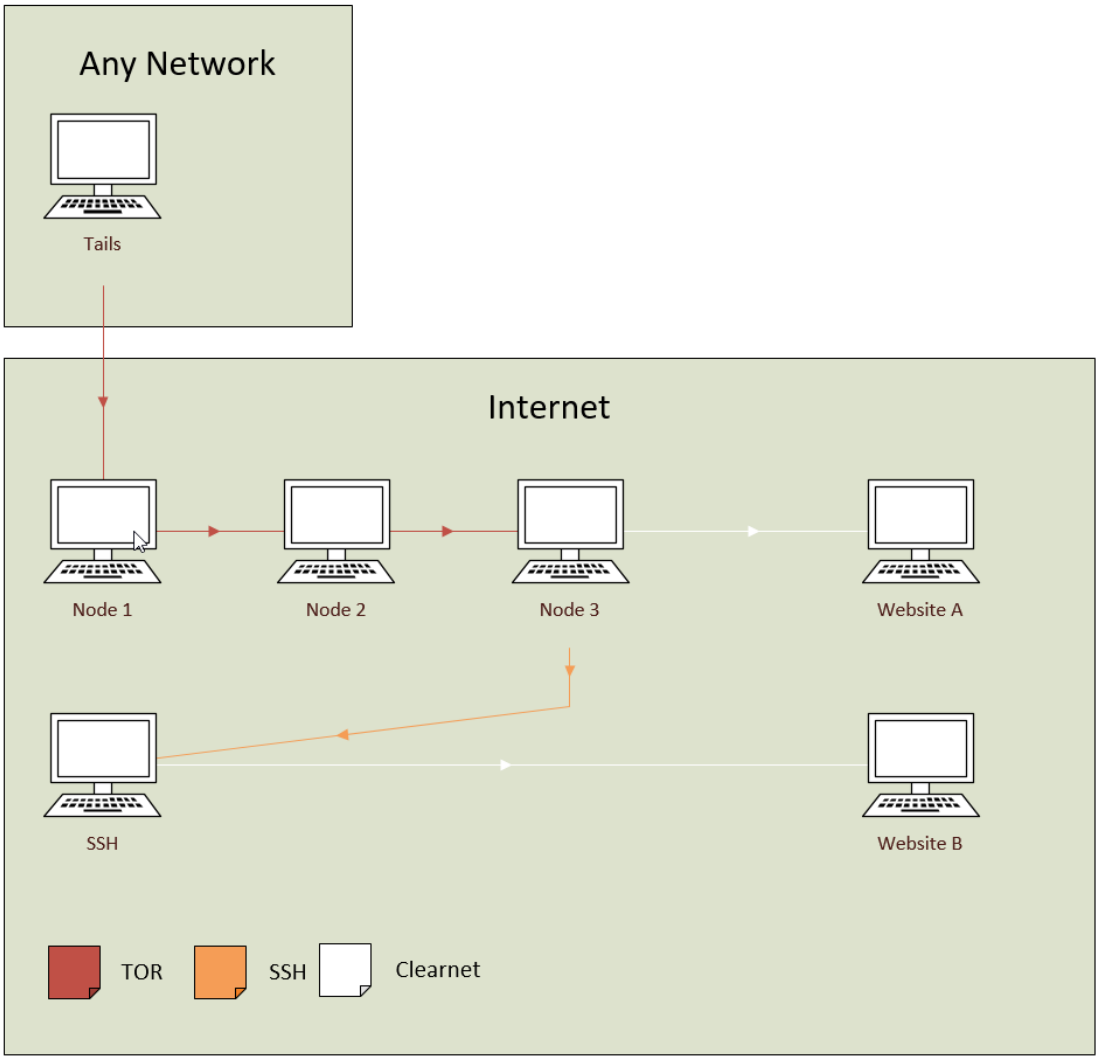
It is vital to emphasize that, as discussed in the previous scenario, all traffic sent via this script can be tracked and analyzed. The http traffic leaves your SSH server in plaintext (unencrypted) before reaching its final destination on the public internet. This means your home Internet Service Provider (ISP) or local government can monitor your activity.

In a Corporate Environment:

If you place a server configured this way within a company network, IT staff or network administrators can also track all traffic generated by your Tails session using this script.

Remember: This monitoring risk only applies to traffic flowing outbound from your SSH server toward the public internet. Traffic routed through the Tor network (crossing the three encrypted nodes) remains secure and encrypted until it exits the final node.

4.0 Using a remote SSH-server anywhere on the Internet (best scenario possible)



Configuring SSH securely, particularly for an Internet-accessible server, can be complex. For users without a second computer on their local network, a suitable solution is to use a third-party SSH provider. This option is often the most practical and reliable way to establish an external SSH connection when using an operating system like Tails with the add-on.

Scenario 03 differs from the previous two scenarios in that your Internet Service Provider (ISP) can only observe encrypted Tor traffic directed to the Internet. Specifically, the ISP can only track the connection made to the first entry node of the Tor circuit. Because all of this traffic is encrypted, the ISP cannot see your online activity.

Every TCP packet you send via Tor passes through three different nodes before reaching its desired destination. The entire communication between the third node (Exit Node) and the foreign SSH server remains encrypted by SSH until the packets leave the SSH server. Across all the scenarios described above, one crucial point should be highly emphasized.

You cannot conceal the use of the Tor Browser or the Tails operating system from your current Internet Service Provider (ISP) using default settings. To hide the fact that you are using Tor or Tails, you must use "bridge mode". This is often the only way for Tails OS to function in highly restrictive environments, such as China. If you are using only the Tor Browser Bundle (not Tails OS) on Windows or Linux within such an environment, alternative methods exist to obfuscate your Tor usage.
In most Western countries, it is not necessary to hide the fact that you are using Tails or the Tor Browser. Internet Service Providers (ISPs) and network operators can observe that users are connecting to the Tor network, but their exact activities remain private. Although they can log connection data, the only specific information visible is the IP address of the first entry node to which the user connects.

There are many cases, such as that of the stupid person who made an "anonymous" bomb threat to Harvard University, highlighting that users sometimes mistakenly believe their actions online are completely untraceable.

https://www.dailydot.com/unclick/tor-harvard-bomb-suspect/

He was apprehended for a simple reason: at the time the threat was made, he was the only individual connected to the campus network using the Tor Browser. Authorities likely speculate he was not using the Tails operating system, which spoofs a computer's real MAC address. By correlating detailed logs from the DHCP server with the specific computer's MAC address, law enforcement easily identified and caught him. Reports at the time indicated the individual faced up to five years in prison and fines of up to $250,000

If you need more specific information about the bridge-mode of Tails, you can visit this following URL from tails.

https://tails.net/doc/anonymous_internet/tor/index.en.html

To find a public SSH server on the Internet, consider the following recommendation and starting points:

https://aruljohn.com/freeshell/
https://www.xshellz.com/
https://freeshell.de/

We re-emphasize that you should only visit these 3 websites using the Tor Browser within the Tails operating system. Accessing them with a standard browser on Linux or Windows to register may compromise your anonymity and fail to hide your personal information effectively.

A great piece of advice from me is to use a fake-email address to register for a SSH-service of your choice. Never use your regular email to register !

I’m sure you will find many SSH-providers on that Websites that meet some or all of your current requirements for a good SSH-provider. And as a third and last piece of important advice from me, never create a username for a login that could be traced back to you. (John Smith / New York)

Some providers only allow the creation of a SOCKS5 proxy and do not offer an interactive shell, such as bash. Many also provide a small amount of space for hosting a simple website. A few include databases like MariaDB or MySQL. The allocated disk space for personal files is often quite limited, sometimes as low as 20 MB or even less. However, I am aware of one SSH provider from that list that offers a generous 40 GB of storage per user.

When using certain SSH servers, a connection attempt made over the Tor (ONION) network, especially through a public exit node, may result in immediate termination. These servers appear to detect and block connections originating from the Tor network, similar to how some websites block Tor traffic.

Most shell-providers do not allow the following “bad things”

- Certain shell providers strictly enforce a limit of exactly one active connection per registered login session. Any attempt to establish a second simultaneous connection using the same credentials is often detected and can result in the user being immediately banned or the connection terminated. This is a policy designed to manage server load and prevent account sharing.

- It is standard practice for the majority of shared service providers, including SSH and shell providers, to strictly prohibit the installation or execution of unauthorized software, which explicitly includes malware.

- It is widely prohibited on most shared server platforms to use popular port-scanning tools like Nmap to scan other servers on the internet. Such activities often violate acceptable use policies (AUPs) because they can be perceived as network reconnaissance or the precursor to an attack, which often leads to immediate suspension of the user's account

- Many service providers strictly forbid the use of Peer-to-Peer (P2P) software, including popular torrent clients. This restriction is primarily due to the high bandwidth consumption associated with P2P traffic, which can strain server resources, and the potential for copyright infringement issues, which can lead to legal complications for the provider.

- It is true that policies regarding the use of Internet Relay Chat (IRC) clients vary among server providers. Some providers permit IRC usage, while many others strictly forbid it due to concerns over bandwidth use or potential misuse. Users should always check the provider's specific terms of service or acceptable use policy to determine whether IRC clients are allowed before attempting to use them.

For experienced Linux administrators who require a high level of trust and customization, purchasing a low-cost Virtual Private Server (VPS) from a provider (such as those found in Europe for around €4 a month) is a viable option. However, this route demands significant expertise. As noted, a single configuration error—especially involving firewall rules—can leave a server highly vulnerable to compromise shortly after deployment. It is not recommended for novices or beginners.

When choosing a VPS provider, experienced users often consider factors like location, price, privacy policies, and available distributions. Legitimate, established providers can be found online. For instance, DigitalOcean offers VPS options, and other services like OVHcloud or Hetzner (which are European companies) provide competitive pricing for virtual servers.
Ultimately, the choice between using a pre-configured shell account and managing a dedicated VPS balances convenience and trust against the necessary technical skill and security responsibility.

It is accurate that successfully securing and managing your own virtual private server requires a specific skill set and motivation. If an individual possesses the required expertise in Linux administration and network security, they can build a highly customized and trusted environment. While this approach carries the risk of configuration errors and potential security breaches if not handled correctly, going down this path undeniably offers significant educational value and provides a deep understanding of both Linux operating systems and security best practices.

5.0 Preparations prior to use of this add-on

To run this script, version 0.90 or higher, you need at least the following items/prerequisites:

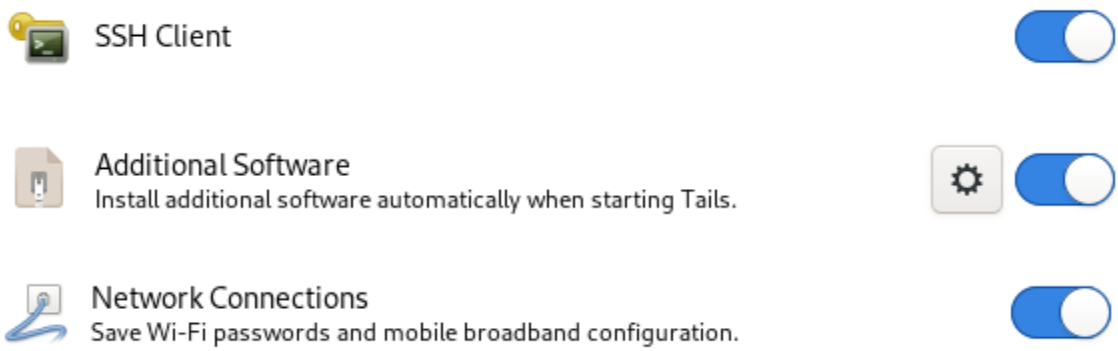
- One USB drive with a minimum capacity of 8 GB is required. For the installation process, please use a very fast drive. In the past, I have used slow drives for Tails, which is something I would never do again.

- A current version of Tails (Version 7.2 or higher) needs to be installed. Tails provides excellent documentation for the installation process itself. For users who need to install Tails, detailed official instructions are available on the Tails operating system official website.

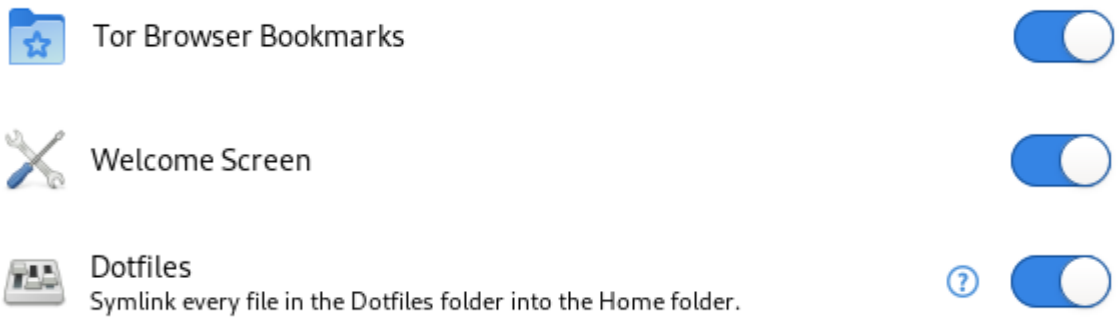
  https://tails.net/install/

- Tails includes a built-in tool for creating an encrypted persistent volume on your USB drive. This allows you to save your personal files, settings, and additional software across different working sessions, which are otherwise securely erased every time Tails shuts down. We look on this topic right now.

The very first thing you must do after the initial boot-up is to create Tails' persistent volume. The add-on cannot run without this persistent volume in place. To be clear, the persistent volume within Tails requires the following three specific options to be activated. These options are mandatory for the add-on to function correctly as expected; they are essential requirements, not merely optional features.



There are three additional options that I highly recommend activating. We could call them “nice to have” options.



The remaining 6 persistent volume options available are :

- Persistent Folder
- Printers
- TOR Bridge
- Electrum Bitcoin Wallet
- GnuPG
- Pidgin Internet Messenger

Whether you use these additional options or not is a matter of personal choice, depending entirely on how you utilize Tails. My add-on is also capable of backing up all the files within a persistent volume, should you wish to do so. From this stage forward, please remember to follow these steps every time you start Tails.

- You have to open the persistent volume on every start of Tails if you want to use the add-on. Of course, you can start Tails without the persistent volume activated, but the add-on itself and all the data that is stored on this persistent volume aren’t accessible, even the stored WiFi passwords aren’t accessible.

- If you are using special or foreign characters in the password for your persistent volume, you must set the correct keyboard layout first. Otherwise, when you type the password in the Tails welcome screen, you will be using the default English (USA) keyboard layout. For users who need to change their keyboard layout, the official Tails documentation explains how to do this: users can change the language and keyboard settings right from the Tails Welcome Screen before logging in.

- Prior to version Tails 4.12, it was not possible to save all settings from the Welcome Screen persistently. By now, this functionality is available, and I personally use it on every Tails OS installation I prepare for myself or someone else.

- If you do not store your Welcome Screen settings inside the persistent volume, you will be required to set the administrator password for Tails manually during every startup. If you omit setting this administrator password, the add-on will be unable to modify the default local firewall. The script makes only one unique, minor adjustment within the Tails firewall configuration. The changed setting of the firewall is very simple.

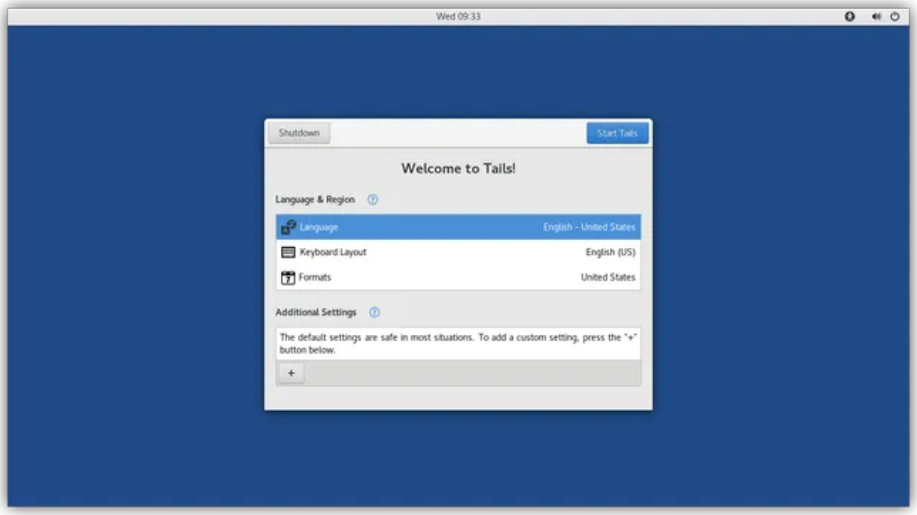
  sudo -S iptables -I OUTPUT -o lo -p tcp --dport 9999 -j ACCEPT

- This minor change allows us to build a local SOCKS5 proxy over SSH. Without this modification, the predefined, very tough default rules from iptables would block any connection attempt made to port 9999 from the internal loopback device 127.0.0.1. Currently, it is not possible to use any other TCP port than 9999 for the SOCKS5 server. In a later release of this add-on, there are plans to make this port freely changeable."

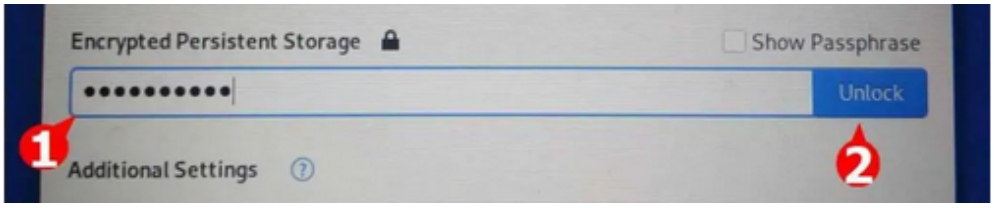
We assume here that you have a working Tails disk and have already created the persistent volume with all the mandatory options.

6.0 Installing the add-on

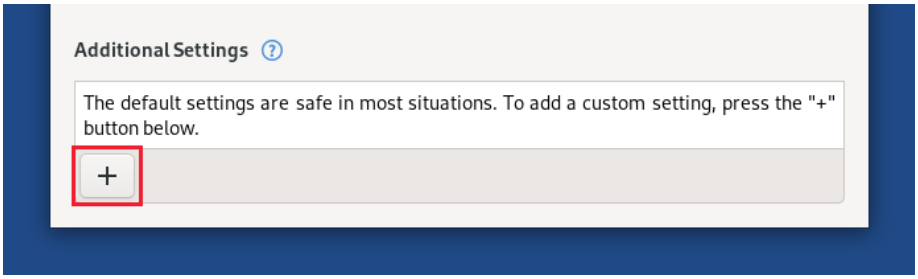
After creating the persistent volume, you must reboot the system to make it active. Following the next startup of Tails, you can change the Language and the Keyboard layout. For users seeking guidance on these steps, official instructions for managing settings in Tails after rebooting are available on the Tails operating system official website.



And then, of course, you activate the persistent volume using the correct password.



And as a final option, you set the administrator password and perhaps disable the 'Unsafe Browser' within the Additional Settings, as I have already mentioned. For users who need to manage these settings, official instructions on configuring administrative passwords and disabling the Unsafe Browser are available in the Tails operating system documentation.



When you activate the 'Greetings' settings for the persistent volume, you only have to do this once. As soon as you unlock the persistent volume during the next startup, all the values you stored there are loaded again, with the exception of the password for the persistent volume itself.

A little tip regarding the keyboard shortcuts on the startup screen of Tails:

Alt+K	Keyboard Layout
Alt+F	Formats
Alt+P	Persistent Storage
Alt+A	Additional Settings
Ctrl+Shift+A	Administration Password
Ctrl+Shift+M	MAC Address Anonymization
Alt+S	Start Tails

For a complete list of navigation shortcuts available on the Welcome Screen, users can consult the Tails operating system official documentation.

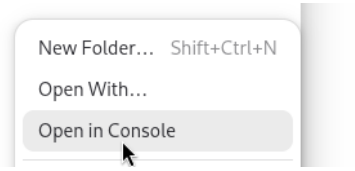
After successfully booting Tails for the second time with the persistent volume unlocked, please open a terminal inside the persistent folder, and then type the following command into the terminal. I will explain to you here how you would do this.

Inside this terminal,you have to type the following Linux command to get the current add-on with the command git.

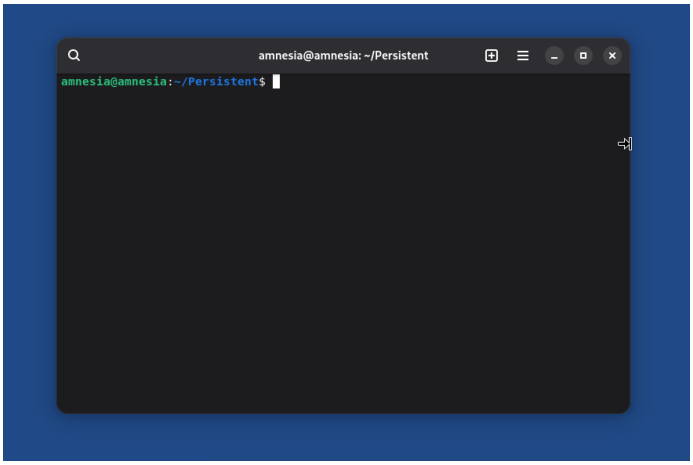
git clone https://github.com/swtor00/swtor-addon-to-tails

For users who need to interact with the terminal in Tails, the official documentation covers fundamental usage and navigation within the system. Information on using the command line can be found in the Tails operating system official documentation. You only need a few commandline commands to make this addon-working. I wonna show you, how you could execute this git commando in a Linux Terminal.

At this early stage, the Persistent volume is empty after the first booting. On the Tails Desktop click on Apps → Favorites → Files. In the new Window click on the left side of “Persistent”. Place the Cursor anywhere to the right Windows and press the right mouse button.



Please activate the Entry : Open in Console → This nice black Window should appear on the Tails Desktop.

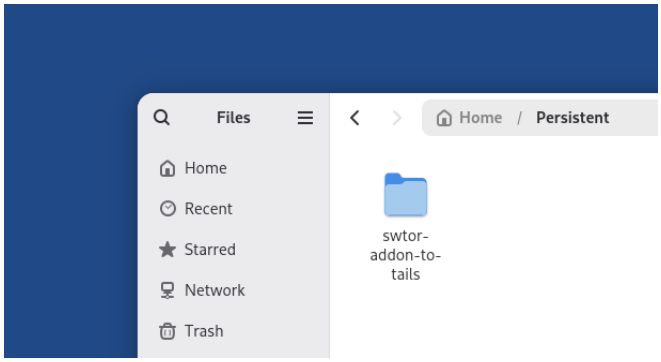


Now type the following command into this Window :

git clone https://github.com/swtor00/swtor-addon-to-tails

If you are finished with the complete command, press <Enter> to execute the command.

You have to wait, until the command git is finished. You can close this window with the command exit and press [enter] to confirm or as a little shortcut with pressing the combination <Ctrl><D>. Please open now the new folder “swtor-addon-to-tails” over the GUI. You should see the content.



Before you do anything inside the add-on's directories, you should first make a few very important decisions about its use. The add-on's complete configuration is written in a single file. You can edit this file using Tails' default editor if you prefer to edit files this way in the terminal. Prior to executing anything, the complete path to this configuration file is as follows.

~/Persistent/swtor-addon-to-tails/swtorcfg/swtor.cfg

The default provided swtor.cfg configuration file (21.11.2025) over github for the add-on version 0.90 looks like this following example.

SWTOR-VERSION:0.90
TAILS-VERSION:7.2
STATE: BETA
HOMEPAGE: https://github.com/swtor00/swtor-addon-to-tails
JOTV :

"No me jodas !"

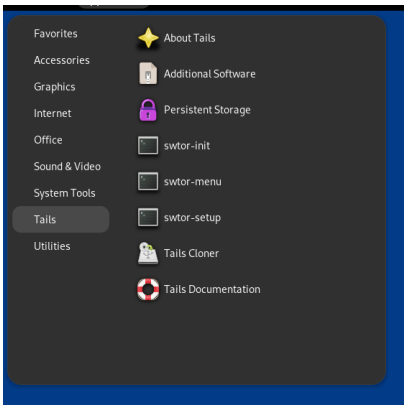
OPTIONS FOR THE SWTOR-ADDON

GUI-LINKS:YES
BROWSER-SOCKS5:YES
BACKUP-FIXED-PROFILE:NO
BACKUP-APT-LIST:NO
TIMEOUT-TB:10
TIMEOUT-SSH:4
TERMINAL-VERBOSE:NO
CHECK-EMPTY-SSH:NO
AUTOCLOSE-BROWSER:YES
XCLOCK-SIZE:180

In the last few lines, you'll see all the relevant entries for the add-on configuration file. All the add-on options are written in CAPITAL letters. Let us now describe the settings that are possible in this file.

GUI-LINKS:YES

If you change the value to NO, you can only start the script via a terminal. Most users should use the predefined default value YES. Prior to Tails Version 6.x, it was possible to link a file directly on the Tails desktop. With the current Tails release 7.2, this is no longer possible. By now, this entry defines whether my script installs three additional entries inside the Tails Application Menu or not, depending on this setting. If you leave it with the default value YES, the Tails menu looks like this example.



If you change it to GUI-LINKS:NO, the Tails menu is not modified and remains untouched as it was. You will then have to execute swtor-menu.sh on every startup, either through the GUI or in a terminal.

BROWSER-SOCKS5:YES

Currently, this entry should always be YES. In the near future, it may be possible to use other settings like RDP-CONNECTION or VNC-CONNECTION.

BACKUP-FIXED-PROFILE:YES
BACKUP-APT-LIST:NO

If you activate both options to the value YES, what exactly does this mean for you as the user of this add-on? In the case that you would like to back up the complete persistent volume, the size of the backup will be around 500 MB or even more. If you leave the values at the default state NO, the created backup will have a size somewhere between 3–10 MB.

TIMEOUT-TB:10

This timeout value in seconds defines how long the scripts should wait before displaying an error message that there is no active internet connection over the onion network. The default value of 10 seconds should work fine for most users of the add-on. If your internet speed is very low, you may have to increase this value to 15 or 20 seconds until you no longer receive connection errors on startup.

TIMEOUT-SSH:4

This timeout value defines how long it takes to successfully connect to a remote SSH server.
If the timeout limit is reached and an active SSH connection has still not been found, a connection error will be displayed.

TERMINAL-VERBOSE:NO

If you would like a verbose output of the shell scripts behind this add-on, you need to set the value to YES.

CHECK-EMPTY-SSH:NO

If you set this option to the state YES, on every startup of the add-on, the script will check for an empty .ssh directory for the user amnesia. If you change this value to YES, the add-on's startup will take a few seconds longer than with the value NO.

AUTOCLOSE-BROWSER:YES

If there is an active SSH connection to a remote host and there are open Chromium browser windows, the script will automatically close all the open browser windows. If you do not like this behavior, change the value to NO and all open Chromium browser instances will remain open.

XCLOCK-SIZE:180

If you are using xclock to display the remote clock from a remote SSH server, this parameter defines the size in pixels. In case your desktop resolution is very high (4k), you may have to increase this value to something like 600 or 700 pixels to see the clock without the help of a binocular.

7.0 Configuring the required SSH-connection for the add-on

For this next configuration step, you will need at least one valid SSH account from anywhere on the Internet or your own personal home server. You may even first try to test this SSH connection with another operating system like Windows, Linux, or an Apple system, which all contain the software needed to establish an SSH connection for testing purposes. Generally, all modern operating systems (including Windows 10 version 1803 or higher) have this SSH software already included.

In my opinion, DO NOT even think about using the scenario discussed above. Under all circumstances, it is an insanely bad idea to use this SSH connection anywhere outside of the Tails system! This is especially true if you are using an SSH server on the Internet !

If you really want to use a SOCKS5 SSH connection to hide your browser traffic from your ISP with any other operating system than Tails, you should create a complete new login on a remote SSH host only for that purpose.

If you were to use these Tails-only SSH credentials outside of Tails, without the protection of the Onion Network in the background, you would leak your currently used WAN IP address to the owner of the remote SSH host immediately the moment you try to connect over SSH. A simple Linux command, who | more inside a terminal of any user, would list all the currently connected users and the corresponding IP address from where they are connected. An example output of the command could look like this:

eao	pts/7	2019-xx-23 17:27 (81.220.101.10)
dyama	pts/8	2019-xx-25 02:07 (158.3.77.185)
tt0077	pts/9	2019-xx-25 20:08 (57.41.129.24)

In the perfect case scenario where we are using the Onion Network inside of Tails to establish the connection to our SSH server, the printed IP address for our own username (column 1) in column 4 would only be from our currently used Exit Node number 3.
Otherwise, it would be our real IP address from our currently used ISP. That would be very bad for the privacy we would like to protect and maintain. Never mix up any SSH login credentials created only for a specific OS. Think about this twice before you do it!

An SSH account with Linux or Unix normally consists of the following information for a successful connection:

- A username like digit1 without any spaces, including a valid password or key files.
- A destination TCP port. The default port for SSH communication is TCP port 22.
- A valid DNS name or an IPv4 IP address to connect to.

All the necessary configuration files for an SSH connection reside inside the directory /home/amnesia/.ssh. If this directory is empty, it means that we have never contacted any SSH server before with our currently activated Tails system. To test our first SSH connection and see if we can successfully log in over SSH, we need to open a terminal in Tails and execute the following command. You can replace the values in this example with the values provided by your own chosen SSH provider.

ssh -p 22 digit1@10.0.1.66

Let us now see how this command is constructed:

ssh: The local Linux command used to communicate encrypted with the remote SSH server.

-p 22: 22 is the default port for an SSH connection. Due to the default behavior of SSH, it is not always necessary to add -p 22 for every connection. If your SSH provider does not use port 22, then the -p XX (replace XX with the port number from your provider) option should be used every time you invoke the command.

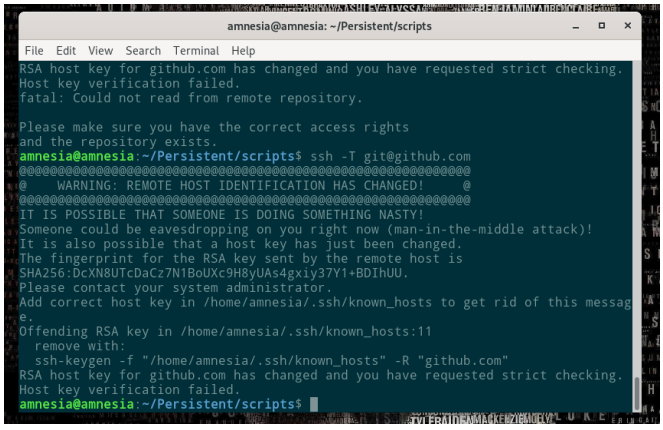
digit1: The username for the SSH connection we are trying to establish.

10.0.1.66: The IP address of the remote server we would like to connect to. We could also provide a user-friendly DNS name instead of an IPv4 IP address. In the following scenario, where we have never made an active SSH connection with our new Tails Medium (including a persistent volume) to an SSH server with the IP address 10.0.1.66, we will see a warning like the following:

The authenticity of host '10.0.1.66 (10.0.1.66)' can't be established. RSA key fingerprint is 90:8c:7a:f8:ae:1a:09:60:44:03:3b:d9:c9:f7:c4:76.
Are you sure you want to continue connecting (yes/no)?

This is the so-called "public fingerprint" of the SSH server we are trying to connect to. The moment we type yes inside the terminal, the public fingerprint for this specific server (10.0.1.66) is then stored inside the file ~/.ssh/known_hosts. After storing this public key inside of Tails, on every subsequent connection we make to the SSH server 10.0.1.66, the already stored value inside the file ~/.ssh/known_hosts will be compared against the value that the server provides upon connecting. If there is a match of both values, we can continue to establish our secure connection.

If the two values do not match, then there is something seriously wrong! If you find entries in your log file or the terminal output like the one shown above, be very careful with your next action. This warning should never be ignored!



https://en.wikipedia.org/wiki/Man-in-the-middle_attack

- If you really decide to make the connection to this host anyway, then someone could now steal your current active password for that particular SSH host, or even worse, steal your currently used public SSH key if you ignore this very important warning and still connect to this possibly evil or nasty SSH server!

- Or it could mean that the public key of the server has been replaced for some natural reason; possibly the remote SSH server was replaced due to a hardware failure and the old, already used SSH keys were never restored.

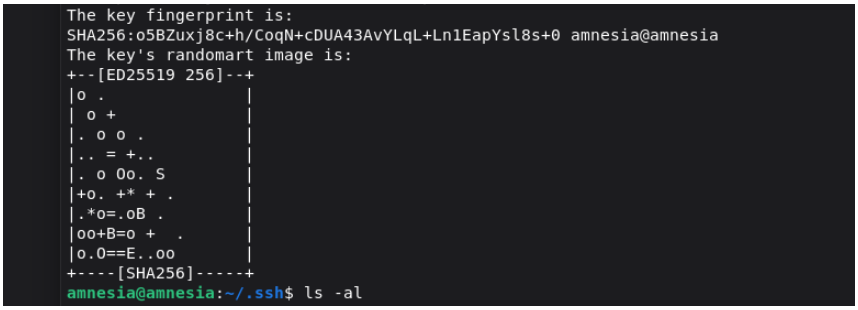
- Next, you should see the password login prompt for the requested user on the remote server. You can now type the password for that account, and the remote shell should appear straight afterwards.

- Please note that you cannot see any character in the password field. Provided we can log in to the remote server without any error, our next step would be to make this login password-free in the future.

Our next terminal command is used to create the private/public key pair for all future SSH communication inside of Tails. The command to accomplish this is the following one:

ssh-keygen -o -a 100 -t ed25519

After a short initialization time to generate these public and private keys, we have an output like the following one. This is now our own personal "holy grail" of encrypted communication for use inside Tails, and it should be saved on a regular basis.



After the creation of our personal SSH keys, we can copy our public key to our SSH server. There is a special SSH command to do this. This copy command will transfer only the public key part to the remote SSH server. The private key part remains in the safe hands of the persistent volume inside Tails

ssh-copy-id -i ~/.ssh/id_ed25519.pub digit1@10.0.1.66

If the SSH server is not using the standard port 22, we have to add the -p X flag (where X stands for the port used by the remote server) in the above example. Some older Unix systems do not support the ssh-copy-id program. In those cases, with a few small bash tricks, it is possible to transfer the public key to the foreign SSH server using standard Unix commands that every system should clearly understand (even the oldest ones).

cd /home/amnesia/.ssh
cat ~/.ssh/id_ed25519.pub | ssh digit1@10.0.1.66 'umask 077; \
cat >> .ssh/authorized_keys’

By now, it should be possible to make this specific SSH connection with Tails to the remote SSH system 10.0.1.66 without any password or any other additional typing on the keyboard. We will now test the SSH connection again by typing the following command.

ssh digit1@10.0.1.66

As you may have noticed, we did not use the flag -p 22 because this is the standard port. A full terminal prompt from the remote host within your current terminal should appear without any password being requested for the connection. For every single SSH host you would like to connect to, you must execute the command ssh-copy-id, or you will have to type the password again on every SSH connection you make! You should test every additional SSH connection carefully so that no confirmation is needed, such as adding the public key to the known_hosts file inside the directory ~/.ssh.

As soon as you have at least one SSH connection that works properly, you can create the configuration file that is needed by the add-on. Inside the doc directory of the add-on, you will find a small example of this configuration file called swtorssh.cfg. All you have to do is edit the file with your own personal values and copy it to the proper location. The configuration of all possible SSH connections that this add-on can manage and use are defined in this single text file. You have to copy the sample file from the doc directory to the correct location or create a new file.

~/Persistent/swtor-addon-to-tails/swtorcfg/swtorssh.cfg

SSH itself is a very complex piece of software. If you would like to have more information about SSH in general or need some cool advanced troubleshooting tips, you should have a closer look by using the Tor Browser to navigate to the following URL.

https://annas-archive.org/md5/a422ecaeb30dab0547ce44134c07d24e

Some smart people call this reference book “The ultimate masterwork of SSH.” If you have any problem related to the complex world of SSH in mind, in this book you will certainly find the answer to all of your current and future questions.

All SSH connections made with the add-on have a "verbose" output on all actions. You will find all the verbose logs of SSH inside the following directory. The verbose mode of all SSH connections inside of this add-on is generated independently from the option TERMINAL-VERBOSE:NO or TERMINAL-VERBOSE:YES. The verbose logs are always generated.

~/Persistent/swtor-addon-to-tails/swtorcfg/log

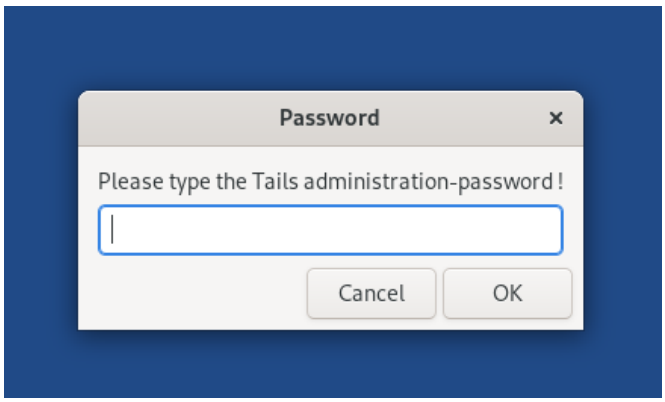
If you have any connection issues, the first thing you should always do is have a closer look at the log files. And please test all SSH connections you would like to use inside of the add-on carefully in a terminal. The SSH connection scripts work very well, but if you make any small mistake when configuring swtorssh.cfg, then the add-on will not work as expected.

8.0 Execute Setup programm for the add-on

For the last step open a Terminal inside ~/Persistent/swtor-addon-to-tails/scripts and execute this command.

./swtor-setup.sh

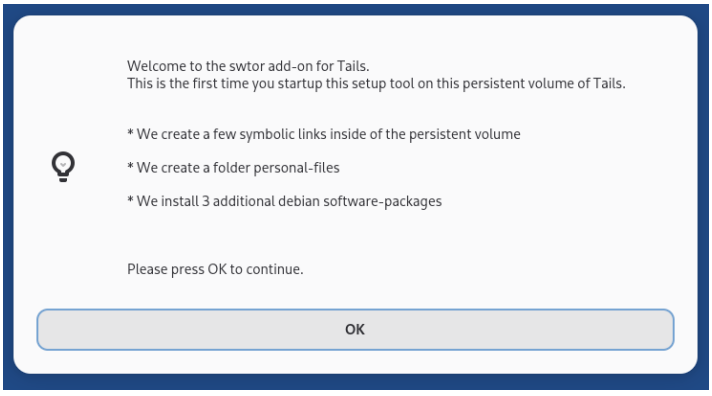
In the background,this script first checks that all mandatory parts of the Persistent Volume are activated. To perform these internal tests,we need the current Administration Password of Tails. If there is no administration password defined, the script will stop here until a password is defined.



Typ it in and press OK. If you miss 3 times the correct password,you have to start over again. The script can now test that :

- The SSH-Client Option is active.
- The Additional Software Option is active.

If there are no errors detected, the next window does appear.



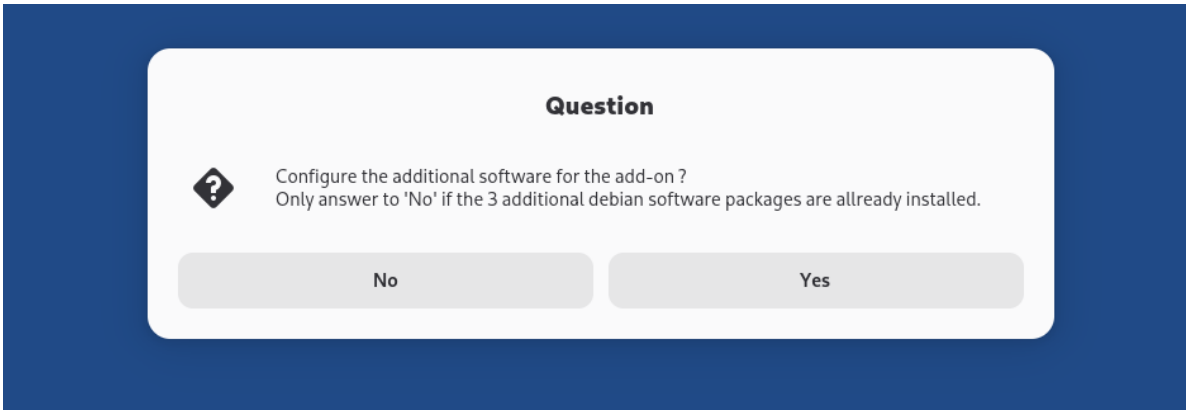
By pressing “OK”, the script creates a few folders automatically inside of the Persistent Volume:

- personal-files inside /home/amnesia/Persistent/personal-files
- doc (links to /home/amnesia/Persistent/swtor-addon-to-tails/doc)
- scripts (links to /home/amnesia/Persistent/swtor-addon-to-tails/scripts)
- settings (links to /home/amnesia/Persistent/swtor-addon-to-tails/settings)
- swtorcfg (links to /home/amnesia/Persistent/swtor-addon-to-tails/swtorcfg)

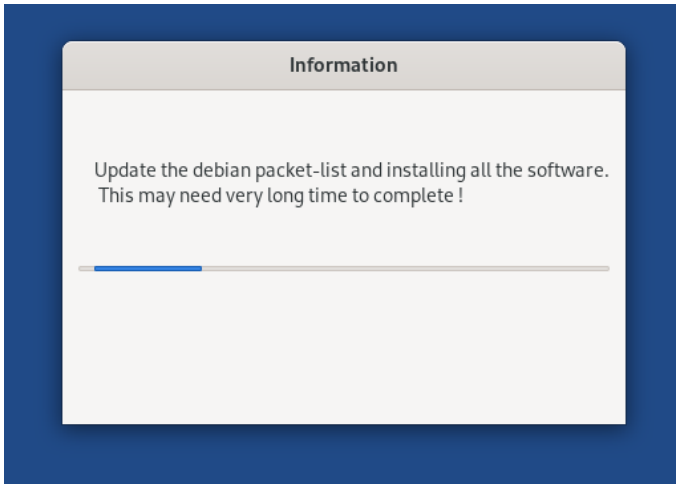
The next step is to install 3 additional Debian packages that have to be installed on every boot with Tails. We need the following additional software in our script.

- chromium
- chromium-sandbox
- sshpass

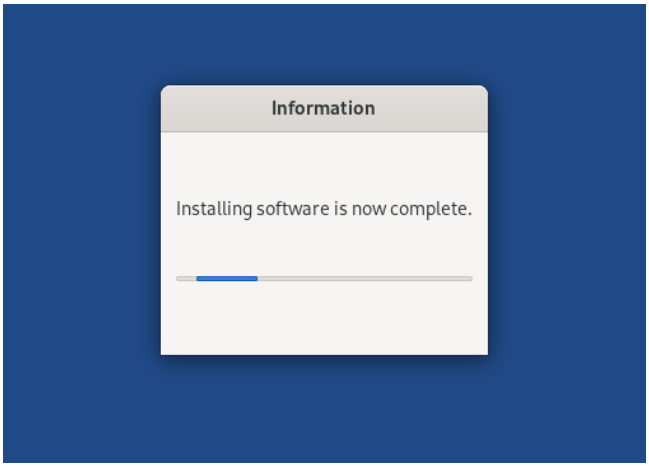
Please confirm the installation of the software in the window you see next.



If you answer "Yes," the three above-named software packages will be installed. You must confirm that all three packages are installed with every boot. The command to install the software may take a long time.



Please remember to mark the three additional software packages as "Install Every Time" when you are given the choice between installing once and every time. As soon as the installation is finished, this window will appear, indicating that the setup of the additional software is complete.



This internal script setup.sh returns the following rc-codes.

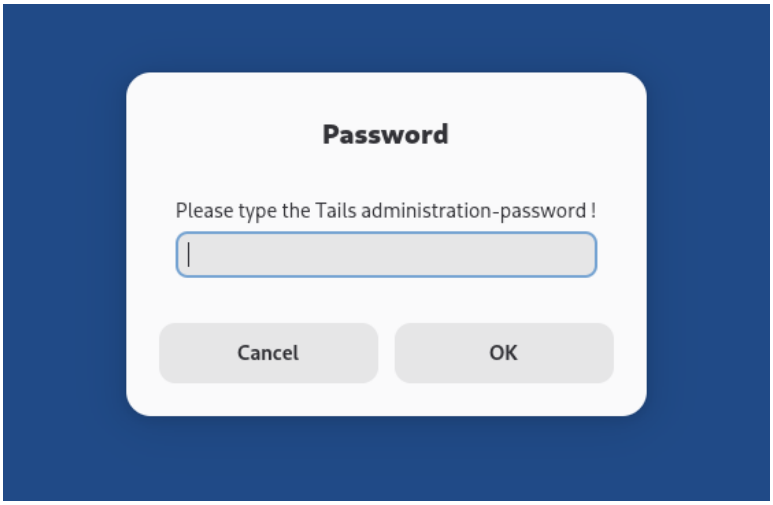
0	no error during execution.
1	error during initialization of swtor-global.sh.
2	Lock directory ~/Persistent/swtor-addon-to-tails/scripts/setup.lock can not be created.
3	setup.sh was already executed. If you would like to execute it again, you have to remove the file ~/Persistent/swtor-addon-to-tails/setup
4	Tails was not started with a administration password. You have to restart Tails and set it.
5	Administration password 3 x times wrong, you have to restart Tails.
6	SSH option is not set. You have to restart Tails and activate this option.
7	The additional Software option is not set. You have to restart Tails and activate this option.

9.0 Execute the add-on for the first time

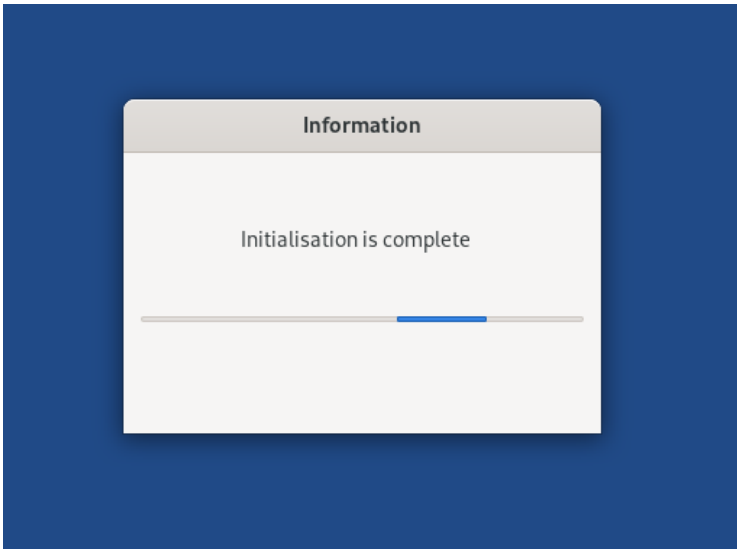
After executing setup.sh, you should perform a reboot. On the next startup of Tails, set the Administration Password and any other desired settings. Navigate to the ~/Persistent/scripts folder. Open a Terminal here, just as you did for the setup.sh script in chapter 8.
Please type the following command:

./swtor-menu.sh

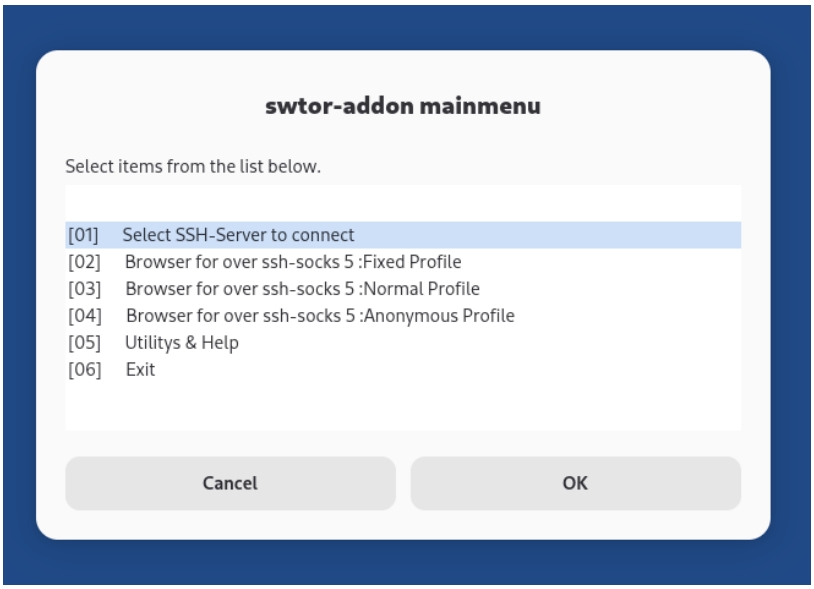
After a few internal checks, the following window will appear, similar to the one you saw during the setup script.



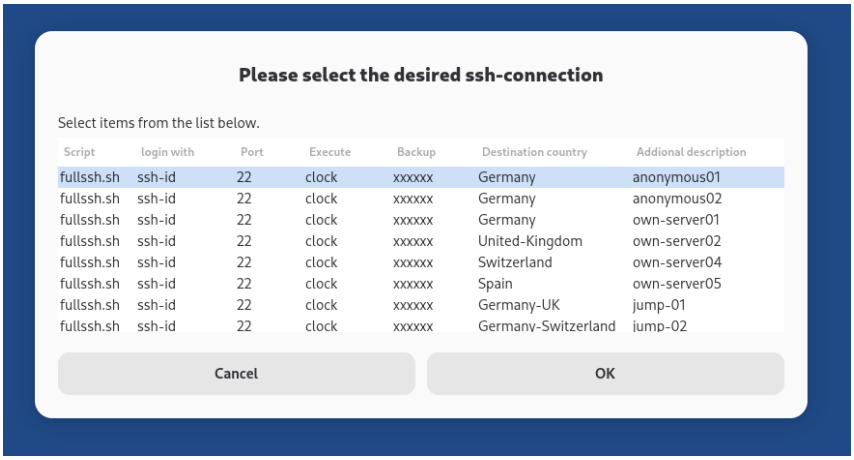
Please type the correct password and wait until you see this message:



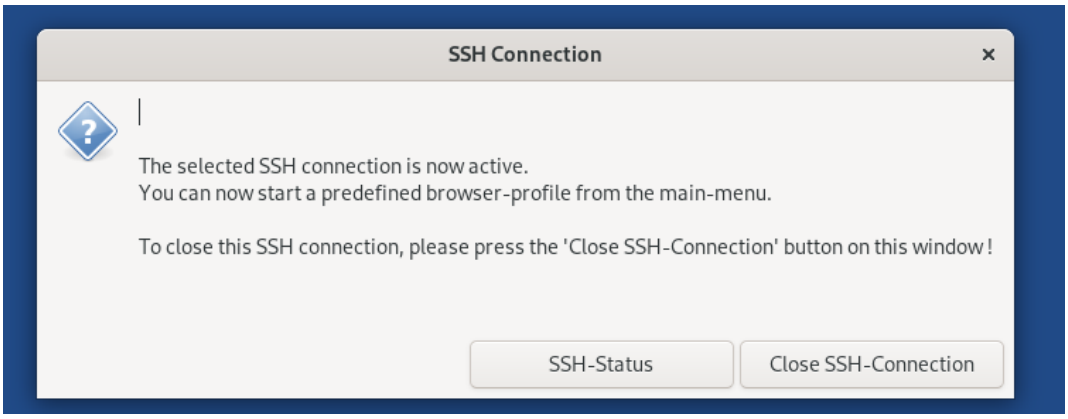
After a few seconds comes the main-screen of the add-on.



After selecting point 01 → and the “OK” Button you should see your current Configuration Screen to select a SSH Server.



For every Server that is defined inside the configuration file you see a single line.
Select any line of the presented configurations an press “OK” to connect to the selected remote Server.
If you have a valid connection without any error this window will appear :

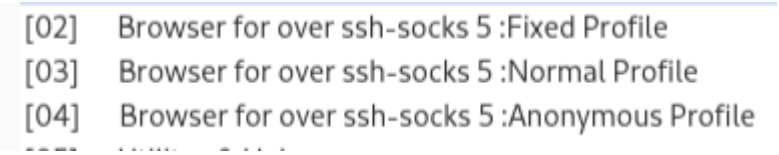


This window will remain open until you press “Close SSH-Connection” or the session ends unexpectedly. If you encounter an error on the selected connection, you will see this and should search for the failure.

But the logs generated inside the directory ~/Persistent/swtorcfg/log/ are often the best help to resolve the problem.

- Did you ever contact the selected SSH server and is a host entry missing ?
- Did you write the password incorrectly ?
- Are the user field and the server written correctly in the configuration file ?
- Is the port number correct ?
- Have you transferred the RSA key if you are using a passwordless login ?

Once the SSH connection is successful, the main menu reappears again, and you can select the Browser Profile you wish to use. It is possible you will only see 2 profiles to select from (Number 03 and Number 04 on your first startup). Fixed Profile (Menupoint 02 → only exist if you have executed a special script inside the scripts directory of the add-on).



Normal Profile (Menupoint 03)

This is the standard profile that most users would select. Please remember that after the connection is closed, all profile data is deleted.

Anonymous Pofile (Menupint 04)

The fundamental purpose of Anonymous Profile is to provide local privacy; it hides your activity from other people who use the same device. It does not make you anonymous online. If you sign in to any website (like Gmail or Facebook) while in Incognito mode, that site will know who you are and can track your activity (including the puplic WAN IP of the SSH-Server). Please remember that after the connection is closed, all profile data is deleted.

Fixed Profile (Menupoint 02)

Any profile data (e.g., passwords and cookies) stored within this profile will remain persistent, even after rebooting Tails. You can even install your own add-ons to this profile, and they will still be available after a reboot. If you would like to create such a profile, you have to execute this script inside the scripts directory.

./cli_create_fixed_profile.sh

This script creates the fixed profile inside the personal-files folder of the Persistent Volume.

~/Persistent/personal-files/3

If you later decide to delete the fixed profile → remove the folder “3” over the GUI or a Terminal.

Warning: With the current SSH connection, you now have the possibility to connect over Tor to websites that would normally block you.

- www.google.com
- www.youtube.com

The two websites mentioned above typically block all attempts to connect over Tor.

10.0 Tools explained → Freezing and Unfreezing

When you open the “Utilities & Help” menu point (05) for the first time, you might ask:
“What are "Freezing" and "Unfreezing" in the add-on ? Why should I use them ?

Let's use a very simple, easy-to-reproduce example.

- Are you tired of the boring blue screen of Tails at startup ?
- Are you tired of starting the add-on on every boot by hand ?
- Would you like the file manager to show hidden files every time you boot ?
- Would you like to use a personal desktop wallpaper ?

Without the so-called “freezing” feature, you would have to set these settings by hand on every boot of Tails. You can use this nice-to-have feature only if your Persistent Volume has “DOT” Files activated. It will not work with deactivated “DOT” files.

You will find an small example file named cli_tweak.sh inside the scripts directory. When you run this script (with ./cli_tweak.sh) , it makes a few small changes to the Tails desktop.

- The desktop wallpaper is now changed.
- Hidden files are shown in the file manager.
- Along with a few other minor adjustments like disabling the webcam and mic.

But these only temporary changes are lost as soon as you make a reboot of Tails. With the help of the freezing feature, you can save your personal configuration even after a reboot of Tails.
If you would like a configuration that you would like to keep after a reboot of Tails, it is easy to achieve this wish. The freezing of a current unfrozen system with all your current settings can be done using the add-on itself or over a terminal. To set the system into the state “Freezed” you have to open a Terminal inside scripts and execute this command.

./cli_freezing.sh

Warning: It is highly recommended that you reboot Tails as soon as possible!

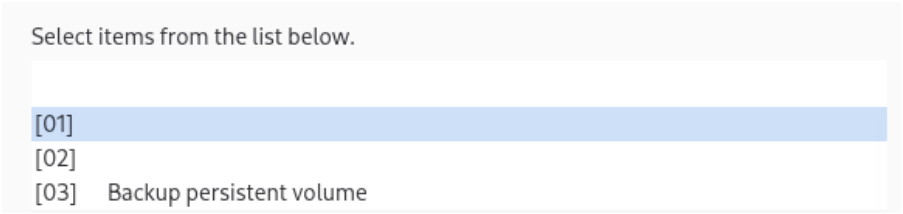
The moment you are booting a frozen Tails, you can undo (unfreeze) all your frozen settings again over the add-on itself or with the terminal.

./cli_unfreezing.sh

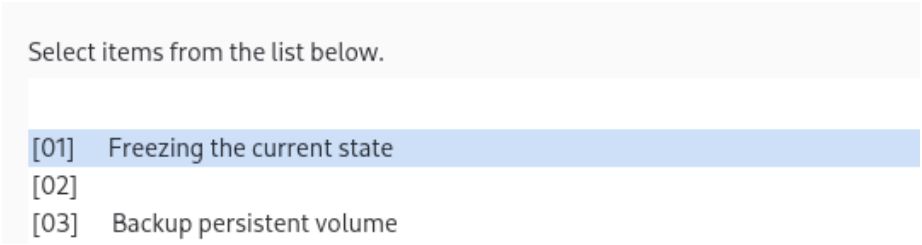
After the next reboot, Tails will return to its default state. Good to know:

If you have the configuration “GUI-LINK:YES” and freeze a Tails system, the add-on will be started automatically in the next startup. And of course, as soon as you unfreeze the system, you have to start it manually on every startup.

Depending on your current active configuration, the Tools menu entries will look different. If your system cannot be frozen (because Dot Files are missing from the persistent volume), the menu will appear as follows:



If your system is in the state, that it can be frozen, it looks like this.



And in the current state “frozen” that it can be “unfrozen” it will looks like this.